

IN THE CLAIMS

Please cancel claims 28 and 40 without prejudice or disclaimer of their underlying subject matter.

Please amend the claims as follows.

23. (currently amended, includes the features of claim 28) A parallel processor comprising, a plurality of processor elements, wherein:

a first processor element of said plurality of processor elements for executing a first user program of a plurality of user programs, said first processor element executes a wait instruction, said wait instruction suspends processing of said first user program; and

a second processor element of said plurality of processor elements for executing a second user program of said plurality of user programs, said second processor element executes a wait release instruction, said wait release instruction commands said first processor element to resume said processing of said first user program,

wherein said first processor element executes a program end instruction, said program end instruction resuming said processing of said second user program.

24. (currently amended, placed into independent form) A parallel processor comprising, a plurality of processor elements, wherein:

a first processor element of said plurality of processor elements for executing a first user program of a plurality of user programs, said first processor element executes a wait instruction, said wait instruction suspends processing of said first user program; and

a second processor element of said plurality of processor elements for executing a second user program of said plurality of user programs, said second processor element executes a wait release instruction, said wait release instruction commands said first processor element to resume said processing of said first user program~~A parallel processor as set forth in claim 23,~~

wherein said second processor element executes a synchronization wait instruction, said synchronization wait instruction suspends processing of said second user program,

said wait instruction suspending processing of said first user program while resuming said processing of said second user program.

25. (previously presented) A parallel processor as set forth in claim 23, wherein plurality of processor elements and a common bus for connecting said plurality of processor elements are installed in a single semiconductor chip.

26. (currently amended, placed into independent form) A parallel processor comprising, a plurality of processor elements, wherein:

a first processor element of said plurality of processor elements for executing a first user program of a plurality of user programs, said first processor element executes a wait instruction, said wait instruction suspends processing of said first user program; and

a second processor element of said plurality of processor elements for executing a second user program of said plurality of user programs, said second processor element executes a wait release instruction, said wait release instruction commands said first processor element to resume said processing of said first user program ~~A parallel processor as set forth in claim 23,~~

wherein said second processor element executes a next instruction without suspending said processing of said second user program after executing said wait release instruction.

27. (previously presented) A parallel processor as set forth in claim 23, wherein said plurality of processor elements perform mutually parallel processing on the basis of instructions written in a program and are capable of communicating with each other via a common bus.

28. (canceled).

29. (previously presented) A parallel processor as set forth in claim 23, further comprising another processor element of said plurality of processor elements for executing another user program, said another processor element executing a program end instruction, said program end instruction resuming said processing of said second user program.

30. (previously presented) A parallel processor as set forth in claim 23, further comprising:

a plurality of local memory, each local memory being uniquely associated with a corresponding processor element of said plurality of processor elements;

a common memory connected to a common bus, said common memory storing said plurality of user programs, a corresponding user program of said plurality of user programs being provided to said corresponding processor element via said common bus.

31. (previously presented) A parallel processor as set forth in claim 30, wherein said first processor element is said corresponding processor element and said first user program is said corresponding user program.

32. (previously presented) A parallel processor as set forth in claim 30, wherein said local memory continues to store said user program until said corresponding processor element executes a program end instruction indicating an end of a program.

33. (previously presented) A parallel processor as set forth in claim 30, wherein, when said second processor element enters a waiting state based on said wait instruction, said corresponding processor element which executed said program execution instruction executes said wait release instruction.

34. (previously presented) A parallel processor as set forth in claim 30, wherein:

said second processor element executes a program execution instruction, said program execution instruction commanding said corresponding processor element to receive said corresponding user program from said common memory and to execute said corresponding user program.

35. (previously presented) A parallel processor as set forth in claim 34, further comprising:

an arbiter for determining which of said of said plurality of processor elements executes a program instructed to be executed by said program execution instruction, and for reading the program instructed to be executed by said program execution instruction from said common memory to said local memory associated with said corresponding processor element.

36. (previously presented) A parallel processing method comprising:

suspending processing of a first user program of a plurality of user programs, said first user program including a wait

instruction, said first processor element executing said wait instruction to suspend said processing of a first user program; and

resuming said processing of said first user program by executing a wait release instruction, said wait release instruction being including within a second user program of a plurality of user programs, a second processor element of said plurality of processor elements for executing said wait release instruction,

said wait release instruction commanding said first processor element to resume said processing of said first user program; and

executing a program end instruction to resume said processing of said second user program, wherein said first processor element executes said program end instruction.

37. (previously presented) A parallel processing method as set forth in claim 36, further comprising:

suspending processing of said second user program, said second processor element executing a synchronization wait instruction.

38. (currently amended, placed into independent form) A parallel processing method comprising:

suspending processing of a first user program of a plurality of user programs, said first user program including a wait instruction, said first processor element executing said wait

instruction to suspend said processing of a first user program;  
and

resuming said processing of said first user program by  
executing a wait release instruction, said wait release  
instruction being including within a second user program of a  
plurality of user programs, a second processor element of said  
plurality of processor elements for executing said wait release  
instruction,

said wait release instruction commanding said first  
processor element to resume said processing of said first user  
program ~~A parallel processing method as set forth in claim 36,~~

wherein in said step of suspending processing of said first  
user program, said processing of said first user program is  
suspended while resuming said processing of said second user  
program.

39. (currently amended, placed into independent form) A  
parallel processing method comprising:

suspending processing of a first user program of a plurality  
of user programs, said first user program including a wait  
instruction, said first processor element executing said wait  
instruction to suspend said processing of a first user program;

resuming said processing of said first user program by  
executing a wait release instruction, said wait release  
instruction being including within a second user program of a  
plurality of user programs, a second processor element of said  
plurality of processor elements for executing said wait release

instruction,

said wait release instruction commanding said first processor element to resume said processing of said first user program ~~A parallel processing method as set forth in claim 36, further comprising;~~ and

executing a next instruction without suspending said processing of said second user program after executing said wait release instruction, said second processor element executing said next instruction.

40. (canceled).

41. (previously presented) A parallel processing method as set forth in claim 36, wherein, when said second processor element enters a waiting state based on said wait instruction, said corresponding processor element which executed said program execution instruction executes said wait release instruction.

42. (previously presented) A parallel processing method as set forth in claim 36, wherein:

said second processor element executes a program execution instruction, said program execution instruction commanding said corresponding processor element to receive said corresponding user program from said common memory and to execute said corresponding user program.



43. (currently amended) A storage medium for storing in a computer-readable format routines comprising:

first processing and second processing to be performed in parallel based on instructions written in programs, wherein

said first processing executes a wait instruction to suspend said first processing by entering said first processing into a waiting state; and

said second processing executes a wait release instruction to resume execution of said first processing,

said second processing enters a synchronization waiting state by executing said wait release instruction until said first processing enters said waiting state when said first processing is not in said waiting state.

44. (previously presented) A storage medium for storing in a computer-readable format routines comprising:

first processing and second processing to be performed in parallel based on instructions written in programs, wherein

said first processing executes a wait instruction to suspend said first processing; and

said second processing executes a wait release instruction to resume execution of said first processing,

said second processing executing a next instruction after executing said wait release instruction without suspending said second processing.